

What is claimed is:

- 1 1. An integrated circuit comprising:
2 a multiplier to produce a product from two floating point multiplicands
3 having a first exponent weight;
4 a floating point conversion unit to convert the product from the first exponent
5 weight to a converted product with a second exponent weight;
6 an adder to produce a present sum from the converted product and a previous
7 sum having the second exponent weight; and
8 a post-normalization unit to convert the present sum to a floating point
9 resultant having the first exponent weight.
- 1 2. The integrated circuit of claim 1 wherein the multiplier is configured to
2 produce a product with an exponent weight of one.
- 1 3. The integrated circuit of claim 2 wherein the floating point conversion unit is
2 configured to convert the product from an exponent weight of one to an exponent
3 weight of thirty-two.
- 1 4. The integrated circuit of claim 1 wherein:
2 the product comprises an exponent having a least significant bit weight of one
3 and a mantissa in carry-save format; and
4 the adder is configured to receive a converted product having an exponent
5 with a least significant bit weight of thirty-two and a mantissa in carry-save format.
- 1 5. The integrated circuit of claim 4 wherein the floating point conversion unit is
2 configured to shift a mantissa of the product by a number of bit positions equal to a
3 value of the least significant five bits of the exponent of the product.

1 6. The integrated circuit of claim 1 wherein the converted product comprises a
2 three bit exponent field having a least significant bit weight of thirty-two.

1 7. The integrated circuit of claim 6 wherein the converted product further
2 comprises a fifty-seven bit mantissa field in carry-save format.

1 8. The integrated circuit of claim 1 wherein the post-normalization unit is
2 configured to be turned off while the adder is producing the present sum.

1 9. A floating point multiply-accumulate circuit comprising:
2 an exponent path including:
3 an exponent summer to sum two input exponents having a first weight
4 to produce a product exponent;
5 an exponent conversion unit coupled to the output of the exponent
6 summer, to convert the product exponent to a second weight;
7 and
8 an exponent accumulation stage to choose a larger exponent from the
9 product exponent and an accumulated exponent; and
10 a mantissa path including:
11 a mantissa multiplier to multiply two input mantissas and produce a
12 product mantissa;
13 a mantissa shifter to shift the product mantissa responsive to the
14 exponent conversion unit in the exponent path; and
15 a mantissa accumulator to accumulate shifted product mantissas.

1 10. The floating point multiply-accumulate circuit of claim 9 wherein the
2 exponent conversion unit is configured to zero the least significant five bits of the
3 product exponent.

- 1 11. The floating point multiply-accumulate circuit of claim 9 wherein the
2 mantissa shifter is configured to shift the product mantissa by a number of bit
3 positions equal to a value of the least significant five bits of the product exponent.
- 1 12. The floating point multiply-accumulate circuit of claim 9 wherein the
2 mantissa accumulator comprises four-to-two compressors.
- 1 13. The floating point multiply-accumulate circuit of claim 9 further comprising
2 a post-normalization stage to produce a normalized floating point resultant.
- 1 14. The floating point multiply-accumulate circuit of claim 13 wherein the post-
2 normalization stage is configured to be turned off until accumulation is complete.
- 1 15. The floating point multiply-accumulate circuit of claim 9 wherein the
2 exponent conversion unit is configured to convert the product exponent to have a
3 least significant bit weight equal to thirty-two.
- 1 16. The floating point multiply-accumulate circuit of claim 9 wherein the product
2 mantissa is in carry-save format.
- 1 17. The floating point multiply-accumulate circuit of claim 16 wherein the
2 mantissa accumulator is configured to accumulate numbers in carry-save format.
- 1 18. A method of performing a multiply-accumulate operation comprising:
2 multiplying two floating point mantissas and summing two floating point
3 exponents to form a product;
4 converting the product to have a different least significant bit weight
5 exponent field;
6 accumulating the converted product; and
7 post-normalizing the accumulated product.

1 19. The method of claim 18 wherein accumulating the product comprises
2 accumulating the product in carry-save format.

1 20. The method of claim 18 wherein accumulating the product comprises adding
2 a first plurality of products with a last product, the method further comprising turning
3 off post-normalization until the last product is accumulated.

1 21. The method of claim 18 wherein converting comprises:
2 shifting a mantissa of the product by an amount equal to the value of the least
3 significant five bits of the exponent of the product; and
4 zeroing the least significant five bits of an exponent of the product.

1 22. The method of claim 18 wherein accumulating comprises:
2 comparing an exponent of a first converted product to an exponent of a
3 second converted product;
4 conditionally shifting right by a fixed amount the mantissa of the converted
5 product having a smaller exponent;
6 selecting the larger exponent as a resultant exponent; and
7 producing a resultant mantissa from a mantissa of the first converted product
8 and a mantissa of the second converted product.

1 23. The method of claim 22 wherein conditionally shifting right comprises
2 selecting one of two inputs of a multiplexor.

1 24. The method of claim 22 wherein producing a resultant mantissa comprises
2 selecting the mantissa of the first converted product if the exponent of the first
3 converted product is more than one greater than the exponent of the second
4 converted product.

1 25. The method of claim 22 wherein producing a resultant mantissa comprises
2 adding mantissas of the first and second converted products to produce a resultant
3 mantissa.

1 26. The method of claim 22 wherein conditionally shifting right comprises:
2 when the exponent of the first converted product is one greater than the
3 exponent of the second converted product, shifting a mantissa of the second
4 converted product thirty-two bit positions to the right.